

智能制造业企业评级系统总测试文档

1 引言

1.1 文档标识

作者	日期	版本
袁易锋	2021-02-19	v1.0

1.2 范围

该文档涵盖了对智能制造业企业评级系统进行的单元测试、集成测试、功能测试的输入及预期结果。

1.3 参考资料

《软件工程与计算（卷二） 软件开发的技术基础》

2 详细情况

2.1 总测试结果概述

依次完成了对智能制造业企业评级系统的单元测试、集成测试、功能测试，测试结果符合预期，产品总体完成企业或个人用户注册登录、更新财务指标、更新两化指标等需求。

2.2 详细测试结果

2.2.1 单元测试

2.2.1.1 测试结果概述

对项目进行了单元测试，测试主要涵盖了dao层的接口，测试确保了dao层的接口规范及逻辑符合预期。

2.2.1.2 单元测试详细结果

测试用例ID	种类	条件	期望结果	测试结果	测试对象ID
TUS1-1	Enterprise	EnterpriseMapper.getEnterpriseById(1);	得到数据库中的id号为1的Enterprise对象	得到Enterprise对象信息	GetEnterpriseById
TUS1-2	Enterprise	EnterpriseMapper.getEnterpriseByEmail("leihuang@163.com");	得到数据库中的email为"leihuang@163.com"的Enterprise对象	得到Enterprise对象信息	GetEnterpriseByEmail
TUS1-3	Enterprise	Enterprise enterprise = new Enterprise(); EnterpriseMapper.insertEnterprise.enterprise);	向数据库中插入一个Enterprise对象信息, 返回1	得到1	InsertEnterprise
TUS1-4	Enterprise	EnterpriseMapper.deleteEnterprise(1);	数据库中删除一个id为1的Enterprise对象, 返回1	得到1	DeleteEnterprise
TUS1-5	Enterprise	Enterprise enterprise = new Enterprise(); EnterpriseMapper.updateEnterprise.enterprise);	数据库中更新Enterprise信息, 返回1	得到1	UpdateEnterprise
TUS1-6	Enterprise	List Enterprises = EnterpriseMapper.getTwentyEnterprises();	得到数据库中至多20个Enterprise对象, 以列表形式返回	得到一个Enterprise列表	GetTwentyEnterprises
TUS1-7	Enterprise	List Enterprises = EnterpriseMapper.getEnterpriseByKey("AI");	得到数据库中关键词有"AI"的Enterprise对象, 以列表形式返回	得到一个Enterprise列表	GetEnterpriseByKey
TUS1-8	Enterprise	List list = new List<>(); EnterpriseMapper.updateEpList(list);	数据库中批量更新Enterprise信息, 返回1	得到1	UpdateEpList
TUS1-9	Enterprise	FinancialIndex financialIndex = new FinancialIndex (); FinanIndMapper.insertFinanInd(financialIndex);	向数据库中插入财务指标, 返回1	得到1	InsertFinanInd
TUS1-10	Enterprise	FinanIndMapper.getFinanByEpld(1);	得到数据库中企业id为1的FinancialIndex经济指标信息	得到FinancialIndex对象	GetFinanByEpld
TUS1-11	Enterprise	List finInds = FinanIndMapper.getAllFinInd();	得到数据库中所有的FinancialIndex对象, 以列表形式返回	得到一个FinancialIndex列表	GetAllFinInd
TUS1-12	Enterprise	FinanIndMapper.deleteFinanById(1);	数据库中删除id为1的FinancialIndex, 返回1	得到1	DeleteFinanById
TUS1-13	Enterprise	MassDiscrete massDiscrete = new MassDiscrete (); MassDisMapper.insertMassDis(massDiscrete);	向数据库中插入离散大批量企业两化指标MassDiscrete, 返回1	得到1	InsertMassDis
TUS1-14	Enterprise	MassDisMapper.getMassDisByEpld(1);	得到数据库中企业id为1的MassDiscrete离散大批量企业两化指标信息	得到MassDiscrete对象	GetMassDisByEpld
TUS1-15	Enterprise	List assDiscrete = MassDisMapper.getAllMassDis();	得到数据库中所有的MassDiscrete对象, 以列表形式返回	得到一个MassDiscrete列表	GetAllMassDis
TUS1-16	Enterprise	MassDisMapper.deleteDiverById(1);	数据库中删除id为1的MassDiscrete, 返回1	得到1	DeleteDiverById
TUS1-17	Enterprise	MixIndustry mixIndustry = new MixIndustry (); MixInduMapper.insertMixIndu(mixIndustry);	向数据库中插入混合行业企业两化指标MixIndustry, 返回1	得到1	InsertMixIndu
TUS1-18	Enterprise	MixInduMapper.getMixIndByEpld(1);	得到数据库中企业id为1的MixIndustry混合行业企业两化指标信息	得到MixIndustry对象	GetMixIndByEpld
TUS1-19	Enterprise	List mixIndustry = MixInduMapper.getAllMixInd();	得到数据库中所有的MixIndustry对象, 以列表形式返回	得到一个MixIndustry列表	GetAllMixInd
TUS1-20	Enterprise	MixInduMapper.deleteDiverById(1);	数据库中删除id为1的MixIndustry, 返回1	得到1	DeleteDiverById
TUS1-21	Enterprise	ProcIndustry procIndustry = new ProcIndustry (); ProcInduMapper.insertProcIndu(procIndustry);	向数据库中插入过程化行业两化指标ProcIndustry, 返回1	得到1	InsertProcIndu

测试用例ID	种类	条件	期望结果	测试结果	测试对象ID
TUS1-22	Enterprise	ProcInduMapper.getProcByEpld(1);	得到数据库中企业id为1的ProcIndustry过程化行业两化指标信息	得到ProcIndustry对象	GetProcByEpld
TUS1-23	Enterprise	List procIndustry = ProcInduMapper.getAllProcInd();	得到数据库中所有的ProcIndustry对象，以列表形式返回	得到一个ProcIndustry列表	GetAllProcInd
TUS1-24	Enterprise	ProcInduMapper.deleteDiverByld(1);	数据库中删除id为1的ProcIndustry，返回1	得到1	DeleteDiverByld
TUS1-25	Enterprise	ServeIndustry serveIndustry = new ServeIndustry (); ServInduMapper.insertServIndu(serveIndustry);	向数据库中插入服务行业两化指标ServeIndustry，返回1	得到1	InsertSerindu
TUS1-26	Enterprise	ServInduMapper.getSerIndByEpld(1);	得到数据库中企业id为1的ServeIndustry服务行业两化指标信息	得到ServeIndustry对象	GetSerIndByEpld
TUS1-27	Enterprise	List serveIndustry = ServInduMapper.getAllSerInd();	得到数据库中所有的ServeIndustry对象，以列表形式返回	得到一个ServeIndustry列表	GetAllSerInd
TUS1-28	Enterprise	ServInduMapper.deleteDiverByld(1);	数据库中删除id为1的ServeIndustry，返回1	得到1	DeleteDiverByld
TUS1-29	Enterprise	SmeDiscrete smeDiscrete = new SmeDiscrete (); SmeDisMapper.insertSmeDis(smeDiscrete);	向数据库中插入中小型离散化行业两化指标SmeDiscrete，返回1	得到1	InsertSmeDis
TUS1-30	Enterprise	SmeDisMapper.getSmeDisByEpld(1);	得到数据库中企业id为1的SmeDiscrete中小型离散化行业两化指标信息	得到SmeDiscrete对象	GetSmeDisByEpld
TUS1-31	Enterprise	List smeDiscrete= SmeDisMapper.getAllSmeDis();	得到数据库中所有的SmeDiscrete对象，以列表形式返回	得到一个SmeDiscrete列表	GetAllSmeDis
TUS1-32	Enterprise	SmeDisMapper.deleteDiverByld(1);	数据库中删除id为1的SmeDiscrete，返回1	得到1	DeleteDiverByld
TUS2-1	User	UserMapper.getUserByld(1);	得到数据库中id为1的User对象	得到Uesr对象	GetUserByld
TUS2-2	User	UserMapper.getUserByEmail("123456@qq.com");	得到数据库中email为"123456@qq.com"的User对象	得到Uesr对象	GetUserByEmail
TUS2-3	User	User user = new User (); UserMapper.createNewUser(user);	创建新的用户，并插入数据库中，返回1	得到1	CreateNewUser
TUS2-4	User	UserMapper.deleteUserByld(1);	数据库中删除id为1的User信息，返回1	得到1	DeleteUserByld
TUS2-5	User	User user = new User (); UserMapper.updateUser(user);	数据库中更新User信息，返回1	得到1	UpdateUser
TUS2-6	User	UserMapper.getUserNum();	获取用户数目	得到用户数目(int)	GetUserNum
TUS2-7	User	Browse browse = new Browse (); BrowseMapper.insertBrowse(browse);	向数据库中插入一条浏览记录，返回1	得到1	InsertBrowse
TUS2-8	User	Browse browse = new Browse (); BrowseMapper.updateBrowse (browse);	数据库中更新浏览记录的时间，返回1	得到1	UpdateBrowse
TUS2-9	User	BrowseMapper.deleteBrowse(1, 1);	数据库中删除userId为1，epld为1的浏览记录，返回1	得到1	DeleteBrowse
TUS2-10	User	List browse = BrowseMapper.selectBrowseByUid (1);	数据库中根据userId获得浏览记录列表	得到Browse的列表	SelectBrowseByUid
TUS2-11	User	BrowseMapper.existBrowse(1, 1);	数据库中得到userId为1，epld为1的浏览记录	得到Browse对象	ExistBrowse

2.2.2 集成测试

2.2.2.1 测试结果概述

在对系统进行单元测试之后，对系统进行的集成测试，包含了对系统service层接口的正确性检验工作。

2.2.2.2 集成测试详细结果

测试用例ID	种类	条件	期望结果	测试结果	测试对象ID
TUS1-1	Enterprise	EnterpriseForm enterpriseForm=new EnterpriseForm(); EnterpriseService.login(userForm);	将前端传入的EnterpriseForm信息与数据库中的Enterprise作密码的比较，正确并返回EnterpriseVO	得到EnterpriseVO	Login
TUS1-2	Enterprise	EnterpriseVO enterpriseVO= new EnterpriseVO(); EnterpriseService.createNewEnterPrise(enterpriseVO);	注册企业，把前端传入的EnterpriseVO信息插入数据库中，返回1	得到1	CreateNewEnterPrise
TUS1-3	Enterprise	EnterpriseService.deleteEnterPrise(1);	数据库删除id为1的Enterprise，返回1	得到1	DeleteEnterPrise
TUS1-4	Enterprise	EnterpriseService.getEnterpriseByld(7);	数据库中得到id为7的Enterprise，返回EnterpriseVO	得到EnterpriseVO	GetEnterpriseByld
TUS1-5	Enterprise	EnterpriseVO enterpriseVO= new EnterpriseVO(); EnterpriseService.updateEnterprise(enterpriseVO);	通过前端传入的新的EnterpriseVO更新数据库中的Enterprise信息，返回1	得到1	UpdateEnterprise
TUS1-6	Enterprise	List enterpriselmgs = EnterpriseService.getEnterpriselmgList();	获取企业注册图片的url，返回List	得到List	GetEnterpriselmgList
TUS1-7	Enterprise	List enterpriseVOs= EnterpriseService.searchEpByKey(String key);	根据前端传入的关键词，在数据库中搜寻匹配的Enterprise，返回List	得到List	SearchEpByKey
TUS1-8	Enterprise	EnterpriseService.updateTable(1);	根据传入的id更新同类型的企业表中的所有评分，此处为1，更新离散大批量企业的评分，无返回值	null	UpdateTable
TUS2-1	User	UserForm userForm= new UserForm(); UserService.login(userForm);	将前端传入的UserForm信息与数据库中的User作密码的比较，正确并返回ResponseVO	得到ResponseVO即User信息	Login
TUS2-2	User	UserVO userVO= new UserVO(); UserService.createUser(userVO);	注册普通用户，把前端传入的UserVO信息插入数据库中，返回ResponseVO	得到ResponseVO，即注册成功的信息	CreateUser
TUS2-3	User	UserVO userVO= new UserVO(); UserService.updateUser(userVO);	通过前端传入的新的UserVO更新数据库中的User信息，返回1	得到1	UpdateUser
TUS2-4	User	UserService.deleteUserByld(2);	数据库删除id为2的User，返回1	得到1	DeleteUserByld
TUS2-5	User	UserService.getUserInfo(9);	数据库中获取id为9的User信息，返回UserVO	得到UserVO	GetUserInfo
TUS2-6	User	UserService.getUserNum();	得到User的数目	得到Integer	GetUserNum

2.2.3 功能测试

2.2.3.1 测试结果概述

系统的功能测试确认观察到的软件的行为遵从软件需求规格说明，并从需求用例的角度来验证功能测试的正确性。

2.2.3.2 功能测试详细结果

测试用例ID	种类	条件	期望结果	测试结果	测试对象ID
TUS1-1	Enterprise	Double[] fin = Double[] fin = {4.0,0.52,2.0,2.5,0.5,2.0,0.05,0.5,1.0,1.0,0.3,0.1,0.5,166.0,2.0,3.0,55.0,1.0}; Double[] div = {95.0,3.0,1.0,-2.0,1.0,0.0,1.0,30.92,2.0,1.0,1.0,8.0,2.0,1.0,2.0,3.0,2.0}; ResponseVO rv = indexController.addEnterpriseTarget(1, div, fin);	往数据库中添加指标，返回 ResponseVO	得到 ResponseVO，即成功添加指标	addEnterpriseTarget
TUS1-2	Enterprise	Double[] fin = {0.0,0.51,2.0,2.5,0.5,2.0,0.05,0.5,1.0,1.0,0.3,0.1,0.5,166.0,2.0,3.0,55.0,1.0}; ResponseVO rv = indexController.updateEnterpriseFinanTarget(2, fin);	更新数据库中id为2的经济指标，返回ResponseVO	得到 ResponseVO，即成功添加指标	updateEnterpriseFinanTarget
TUS1-3	Enterprise	Double[] div = {50.0,4.0,2.0,-5.0,5.0,3.0,3.0,39.43,9.0,1.0,5.0,65.0,3.0,1.0,4.0,4.0,4.0}; ResponseVO rv = indexController.updateEnterpriseDiverTarget(2, div);	更新数据库中id为2的两化指标，返回ResponseVO	得到 ResponseVO，即成功添加指标	updateEnterpriseDiverTarget
TUS1-4	Enterprise	ResponseVO responseVO = indexController.getEnterpriseTarget(1);	数据库中得到id为1的企业的指标，返回ResponseVO	得到指标	getEnterpriseTarget
TUS2-1	User	UserVO userVO = new UserVO(); BeanUtils.copyProperties(user, userVO); UserForm userForm = new UserForm(); userForm.setEmail(user.getEmail()); userForm.setPassword(user.getPassword()); ResponseVO result = userController.login(userForm);	将前端传入的 UserVO中的信息与数据库中的 User信息比较，返回ResponseVO	得到User信息	Login
TUS2-2	User	UserVO userVO = new UserVO(); ResponseVO result = userController.register(userVO);	将前端传入的 UserVO信息插入数据库中，返回 ResponseVO	插入成功	register
TUS2-3	User	UserVO userVO = new UserVO(); ResponseVO result = userController.updateUser(userVO);	将前端传入的 UserVO信息更新到数据库中，返回 ResponseVO	更新成功	updateUser
TUS2-4	User	ResponseVO result = userController.deleteUser(2);	数据库中删除id为2的User，返回 ResponseVO	删除成功	deleteUser
TUS2-5	User	ResponseVO result = userController.getUserInfo(4);	数据库中获取id为4的User，返回 ResponseVO	获取成功	getUserInfo
TUS2-6	User				

3 其他

· 详细测试用例代码参考 项目文件里的test文件。